***Overall Goal***

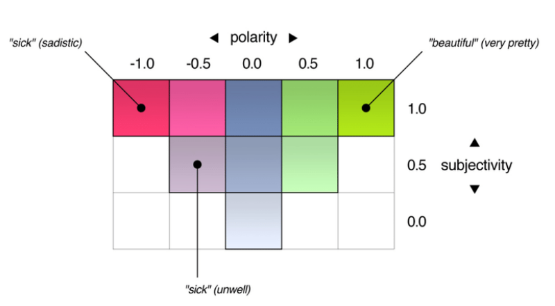
*To build a web application which can scan content online (e.g. news articles) and provide insights into the intent/sentiment behind them. In the world of mass media, it can be difficult to remain aware of the validity or intent behind the content you are consuming.*

*As a User I want to be able to:*

*• Input a website URL or upload a document to be reviewed*

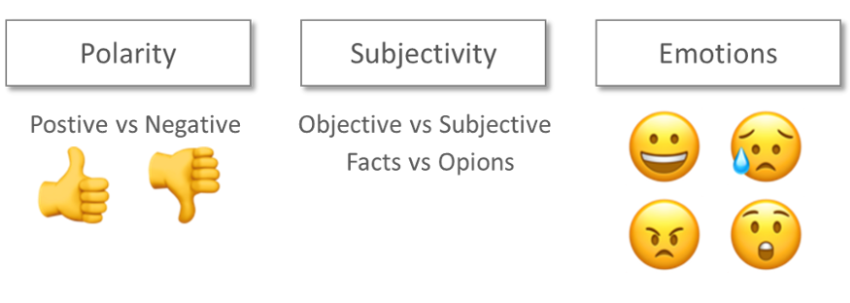
*• View the output analysis*

1. INPUT AN URL => Used Rule Based Approach for Sentiment Analysis (not ML)
   1. TextBlob = polarity and subjectivity
   2. Vader = only polarity, more applicable for social media (not applicable)
2. OUTPUT ANALYSYS
   1. Sentiment (polarity / subjectivity)
   2. Credibility (Score, Category and Source)
   3. Others (future):
      1. Top Image
      2. Keywords
      3. Summary



<https://towardsdatascience.com/sentiment-analysis-and-the-moon-landing-2466baf65a53>

Sentiment analysis covers different categories such as polarity, subjectivity and even emotions such as happiness, sadness, anger and surprise:

**Rule based vs machine learning**

There are two main approaches to do sentiment analysis:

* **Rule-based expert systems (a.k.a. lexicon approach)**
* Machine learning (subfield of Artificial Intelligence)

The rule-based method relies on a ‘lexicon’, which is a large vocabulary of words with a predefined sentiment value for each word. For instance, the word ‘hate’ gets a sentiment -0.57 compared to ‘love’ of +0.63. In its simplest form the algorithm adds up all the sentiment values for each word in a text in order to calculate an overall sentiment. The advantages of rule-based methods are that given a large enough lexicon they are fairly accurate, and one can easily describe how each sentiment classification where made.

The second approach is based on machine learning, which like humans learn from experience. In general, the more experiences or training that are performed the better these models are at predicting sentiments. There are numerous machine learning algorithms, and one of the tasks for a data scientist is to understand which ones are relevant for which type of problem. The machine learning approach has the potential to scale better than the rules-based approaches and has proven to be surprisingly accurate. However, it requires large amount of training data and for some algorithms it is harder to explain how a certain prediction was made, compared with the more transparent rule-based approach. We will take a closer at how we can apply both machine learning and rule-based methods to do sentiment analysis and compare them. A third option, commonly used in industry, is to use a hybrid approach combining the best of both worlds.

Rule-based models (no training needed)

As mentioned previously we want to evaluate the accuracy of rule-based and machine learning models. For the rule based evaluation we will look into an open source framework VADER (Valence Aware Dictionary and sEntiment Reasoner). Vader (lets stick with lower-case) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. Through the rules engine Vader “listens” to modifying words (strengthen/weakening), use of capital letters and exclamation marks (strengthening the sentiment). Vader even has support for emojis and emoticons.

VADER

<https://thinkinfi.com/sentiment-analysis-using-vader-in-python/>

If you don’t have training data for sentiment analysis you must use rule based approach. You can make your own rule or you can simply use tool like VADER. There are multiple tools for sentiment analysis available like: StanfordCore NLP, TextBLOB etc.

SPACY

<https://spacy.io/usage/spacy-101>

spaCy’s models are **statistical** and every “decision” they make – for example, which part-of-speech tag to assign, or whether a word is a named entity – is a **prediction**. This prediction is based on the examples the model has seen during **training**. To train a model, you first need training data – examples of text, and the labels you want the model to predict. This could be a part-of-speech tag, a named entity or any other information.

This also means that in order to know how the model is performing, and whether it’s learning the right things, you don’t only need **training data** – you’ll also need **evaluation data**. If you only test the model with the data it was trained on, you’ll have no idea how well it’s generalizing. If you want to train a model from scratch, you usually need at least a few hundred examples for both training and evaluation. To update an existing model, you can already achieve decent results with very few examples – as long as they’re representative.

Rule-based matching

Find phrases and tokens, and match entities

Compared to using regular expressions on raw text, spaCy’s rule-based matcher engines and components not only let you find the words and phrases you’re looking for – they also give you access to the tokens within the document and their relationships. This means you can easily access and analyze the surrounding tokens, merge spans into single tokens or add entries to the named entities in doc.ents.

SPACY Rule-based Matching

1. PhraseMatcher (Spacy.matcher) = checks if a specific set of words can be found in the text
2. SpaCy's English models were trained on defines a PERSON entity as just the person name
   1. Spacy english models = Available pretrained statistical models for English (eg en\_core\_web\_sm, en\_core\_web\_md, en\_core\_web\_lg)

import spacy

nlp = spacy.load("en\_core\_web\_sm")

doc = nlp("Dr. Alex Smith chaired first board meeting of Acme Corp Inc.")

print([(ent.text, ent.label\_) for ent in doc.ents])

==>> [('Alex Smith', 'PERSON'), ('first', 'ORDINAL'), ('Acme Corp Inc.', 'ORG')]

\*\*\*\* Highlighting “Interesting” words or statements

<https://spacy.io/api/annotation>

## **NER -** [**Named Entity Recognition**](https://spacy.io/api/annotation#named-entities)

|  |  |
| --- | --- |
| **TYPE** | **DESCRIPTION** |
| PERSON | People, including fictional. |
| NORP | Nationalities or religious or political groups. |
| FAC | Buildings, airports, highways, bridges, etc. |
| ORG | Companies, agencies, institutions, etc. |
| GPE | Countries, cities, states. |
| LOC | Non-GPE locations, mountain ranges, bodies of water. |
| PRODUCT | Objects, vehicles, foods, etc. (Not services.) |
| EVENT | Named hurricanes, battles, wars, sports events, etc. |
| WORK\_OF\_ART | Titles of books, songs, etc. |
| LAW | Named documents made into laws. |
| LANGUAGE | Any named language. |
| DATE | Absolute or relative dates or periods. |
| TIME | Times smaller than a day. |
| PERCENT | Percentage, including ”%“. |
| MONEY | Monetary values, including unit. |
| QUANTITY | Measurements, as of weight or distance. |
| ORDINAL | “first”, “second”, etc. |
| CARDINAL | Numerals that do not fall under another type. |

<https://www.quora.com/How-can-I-use-Spacy-to-perform-sentiment-analysis>

Spacy doesn’t have a pre-created sentiment analysis model.